Geometric Algebra - The mathematical language for Computational Engineering?

Dietmar Hildenbrand[1]

1) University of Technology Darmstadt
dietmar.hildenbrand@gris.informatik.tu-darmstadt.de

Abstract This work reviews some current engineering applications of geometric algebra and observes the potential of this mathematical language to become a basis for a wide range of computational engineering applications. Geometric algebra unifies many other mathematical concepts like quaternions and projective geometry and is able to easily deal with geometric objects, operations and transformations. For computational engineering, not only these nice properties are important, but also the computational point-of-view. We describe the big potential of geometric algebra for optimizations leading to highly efficient implementations.

1 Introduction

Geometric algebra is a mathematical framework to easily describe geometric aspects. It allows us to develop algorithms fast and in an intuitive way. Geometric algebra is based on the work of Hermann Grassmann and William Clifford ([12], [13]). Pioneering work has been done by David Hestenes, who firstly applied geometric algebra to problems in mechanics and physics [23] [22].

2 Benefits of geometric algebra for computational engineering

Here, we highlight some properties of geometric algebra making it very helpful for computational engineering applications.

2.1 Unification of mathematical systems

In the wide range of engineering applications a lot of different mathematical systems are used. One big advantage of geometric algebra is, that it is able to cover a lot of other mathematical systems like vector algebra, imaginary numbers, quaternions, Plucker coordinates or dual quaternions. Some of the applications described in section 3 are using this property very advantageously.

2.2 Easy handling of geometric objects

Geometric algebra is able to easily treat geometric objects like spheres, circles and planes as well as geometric operations between them. The spheres of Fig.

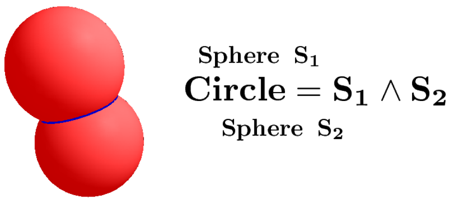

Fig. 1. Spheres and circles are basic entities of geometric algebra. Geometric operations like the intersection of two spheres can be expressed very easy.

1 for instance are simply represented by the algebraic object

$$S = P - \frac{1}{2}r^2 e_\infty \tag{1}$$

based on their center point P, their radius r and the basis vector e_∞ representing the point at infinity. Their intersecting circle can be computed easily with the help of the outer product as a basic algebraic operation

$$Z = S_1 \wedge S_2 \tag{2}$$

This way of computing with geometric algebra can be advantageously used in application areas like robotics, computer vision and computer graphics as presented in section 3.

2.3 Easy handling of rigid body motion

Rigid body motions can be described with the help of one compact linear expression, called screw, including both the rotational and the linear part. In the same way velocities and forces are represented. The combinations of rotational and linear velocities as well as of forces and torques are also described with the help of one linear expression including both the rotational and the linear parts. One application of this property is the improvement of Finite Element methods by geometric algebra [7].

2.4 Easy handling of curl, vorticity and rotation

Addressing the curl, vorticity and rotation, geometric algebra offers some advantages over "classical" vector algebra. For instance, the outer product allows for the curl to be defined in any dimension, where the cross-product is only defined for three-dimensional vectors. This kind of properties based on geometric algebra calculus opens the door for applications e.g. for fluid dynamics [11].

2.5 Potential for very efficient implementations

Geometric algebra is a very powerful tool to solve engineering problems. But, often a clear structure and greater elegance results in lower runtime performance. In section 4 we will present our optimization approach. We will see there that geometric algebra inherently has a big potential for optimizations leading to highly efficient implementations.

3 Current applications

The research and application focus of geometric algebra in the engineering area currently is on computer graphics, robotics and computer vision. This is why we describe these applications in some more detail.

Since about one decade, many researchers at the University of Cambridge, UK have shown that applying geometric algebra in their field of research is very advantageous. They started with projects more related to computer vision. In the meantime the Cambridge engineering department as well as a company with university background are dealing with typical computer graphics applications like mesh deformation and lighting. Lasenby et al. and Perwass et al. present some applications dealing with structure and motion estimation as well as with the trifocal tensor in the articles [27], [28] and [37, 38, 36]. Some computer graphics articles using geometric algebra are presented by Cameron et al. [10] and Wareham et al. [46], [47]. They use geometric algebra for applications like rigid-body pose and position interpolation, mesh deformation and catadioptric cameras. Geomerics [1] is a start-up company in Cambridge specializing in simulation software for physics and lighting which just presented its new technology allowing real-time radiosity in videogames utilizing commodity graphics processing hardware. The technology is based on geometric algebra wavelet technology.

Dorst et al. at the University of Amsterdam, the Netherlands, are applying their fundamental research on geometric algebra [18, 15, 17, 29, 30] to computer graphics. Zaharia et al. investigated modeling and visualization of 3D polygonal mesh surfaces using geometric algebra [48]. Currently D. Fontijne is primarily focusing on the efficient implementation of geometric algebra. He investigated the performance and elegance of five models of 3D Euclidean geometry in a ray tracing application [21] and developed a code generator for geometric algebras [20]. There is a book with applications of geometric algebra edited by Dorst et al. [14]. A new book was published recently [16].

The first time geometric algebra was introduced to a wider Computer Graphics audience, was probably at the SIGGRAPH conferences 2000 and 2001 (see [31]).

Bayro-Corrochano et al. from Guadalajara, Mexico are primarily dealing with the application of geometric algebra in the field of robotics and computer vision. Some of their kinematics algorithms can be found in [3] for the 4D motor algebra and in the Conformal geometric algebra papers [5, 6] dealing with inverse kinematics, fixation and grasping as well as with kinematics and differential kinematics of binocular robot heads. Books from Bayro-Corrochano et al. with geometric algebra applications are for instance [2] and [4].

At the University of Kiel, Germany, Sommer et al. are applying geometric algebra to robot vision [44], e.g. Rosenhahn et al. concerning pose estimation [41, 42] and Sommer et al. regarding the twist representation of free-form objects [45]. Perwass et al. are applying conformal geometric algebra to uncertain geometry with circles, spheres and conics [33], to geometry and kinematics with uncertain data [34] or concerning the inversion camera model [35]. There is a book with applications of geometric algebra edited by Sommer [43] and a new book about the application of geometric algebra in engineering applications by Christian Perwass [39]. Sven Buchholz together with Kanta Tachibana from the university of Nagoya and Eckhard Hitzer from the university of Fukui, Japan do some interesting research dealing for instance with neural networks based on geometric algebra ([9], [8]).

In addition to these examples there are a lot of other applications like geometric algebra fourier transform for the visualization and analysis of vector fields [19] or classification and clustering of spatial patterns with geometric algebra [40] showing the wide area of possibilities of advantageously using this mathematical system in engineering applications.

4 Computational efficiency of geometric algebra algorithms with the help of Gaalop

Gaalop uses a two-stage approach for the automatic optimization of geometric algebra algorithms. In a first step we optimize our geometric algebra algorithm with the help of symbolic computing. This kind of optimization results in very basic algorithms leading to high efficient software implementations. These algorithms which are in a first step optimized have an inherent high potential of parallelization which can be used very advantageously for hardware optimizations in a second step.

We investigated these performance issues especially with an inverse kinematics algorithm. Naively implemented, our first algorithm was slower than the conventional one. But, with our symbolic computation optimization approach the software implementation became three times faster [24] and a hardware implementation about 300 times faster [25] (3 times by software optimization and 100 times by additional hardware optimization) than the conventional software implementation. This result served us as a proof-of-concept for Gaalop [26].

Figure 2 shows an overview over the architecture of Gaalop. Its input is a geometric algebra algorithm. Via symbolic simplification it is transformed into a generic intermediate representation (IR) that can be used for the generation of different output formats such as C-code, FPGA descriptions (as a structural hardware description, currently written in the Verilog language), CLUCalc-code in order to visualize the results (see [32]).

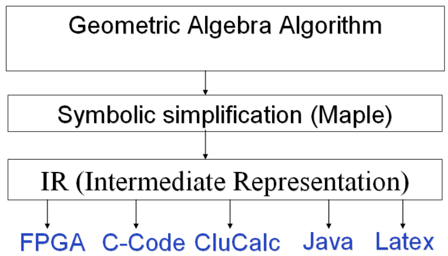

Fig. 2. Architecture of Gaalop

Gaalop is currently able to handle sequential geometric algebra algorithms. It optimizes parts of the algorithm that are explicitly indicated. The algorithm is currently transformed into C-code as well as CLUCalc-code. Please find always the newest information on the Gaalop homepage ([26]).

In the future we will be able to handle not only sequential algorithms but also more advanced program structures. One focus will lie on the automatic compilation of hardware descriptions as well as on mixed solutions handling reasonable combinations of software and hardware implementations.

From the runtime performance point-of-view, Gaalop is just able to automatically generate software implementations comparable to the software implementation of our proof-of-concept application ([25]). In this application we are three times faster than the conventional algorithm (see [24]). In the future, we expect an additional speedup of more than 100 times based on the automatic generation of hardware implementations as described in [25].

5 Conclusion

In this paper, we highlight some properties of geometric algebra that are already very helpful in a lot of presented engineering applications. With these properties together with the potential of being the base for highly efficient implementations using tools like Gaalop we are convinced that geometric algebra will become more and more fruitful in a great variety of computational engineering applications.

As a consequence, what are the benefits for students, researchers and practitioners with geometric algebra? From the educational point of view, students do not have to learn different mathematical systems and the translations between them, but one global mathematical system. Researchers can get new insights into their research area using this global system. Practitioners in the field of computational engineering can benefit from the easy development, test and maintenance of algorithms based on geometric algebra.

References

1. The homepage of geomerics ltd. HTML document http://www.geomerics.com.
2. E. Bayro-Corrochano. Robot perception and action using conformal geometry. In E. Bayro-Corrochano, editor, *the Handbook of Geometric Computing. Applications in Pattern Recognition, Computer Vision, Neurocomputing and Robotics*, chapter 13, pages 405–458. Springer Verlag, Heidelberg, 2005.
3. E. Bayro-Corrochano, K. Daniilidis, and G. Sommer. Motor algebra for 3d kinematics : The case of the hand-eye calibration. *Journal of Mathematical Imaging and Vision*, 13:79–99, 2000.
4. E. Bayro-Corrochano and G. Sobczyk, editors. *Geometric Algebra with Applications in Science and Engineering.* Birkhäuser, 2001.
5. E. Bayro-Corrochano and J. Zamora-Esquivel. Inverse kinematics, fixation and grasping using conformal geometric algebra. In *IROS 2004, September 2004, Sendai, Japan*, 2004.
6. E. Bayro-Corrochano and J. Zamora-Esquivel. Kinematics and differential kinematics of binocular robot heads. In *proceedings of ICRA conference, Orlando, USA*, 2006.
7. E. Brendel, Thomas Kalbe, Dietmar Hildenbrand, and Michael Schfer. Simulation of elastic rods using conformal geometric algebra. In *International Symposium on Frontiers of Computational Science, Nagoya, Japan*, 2008.
8. S. Buchholz, E. Hitzer, and K. Tachibana. Coordinate independent update formulas for versor clifford neurons. In *Proc. Joint 4th International Conference on Soft Computing and Intelligent Systems and 9th International Symposium on advanced Intelligent Systems (SCIS and ISIS 2008)*, Nagoya, Japan, 17-21 September 2008.
9. S. Buchholz, K. Tachibana, and E. Hitzer. Optimal learning rates for clifford neurons. In *International Conference on Artificial Neural Networks*, volume 1, pages 864–873, Porto, Portugal, 9-13 September 2007.
10. Jonathan Cameron and Joan Lasenby. Oriented conformal geometric algebra. *Proceedings of ICCA7*, 2005.
11. Carsten Cibura. Derivation of fluid dynamics basics using geometric algebra. Master's thesis, TU Darmstadt, 2007.
12. William K. Clifford. Applications of grassmann's extensive algebra. In R. Tucker, editor, *Mathematical Papers*, pages 266–276. Macmillian, London, 1882.

13. William K. Clifford. On the classification of geometric algebras. In R. Tucker, editor, *Mathematical Papers*, pages 397–401. Macmillian, London, 1882.
14. L. Dorst, C. Doran, and J. Lasenby, editors. *Applications of Geometric Algebra in Computer Science and Engineering*. Birkhäuser, 2002.
15. L. Dorst and D. Fontijne. 3d euclidean geometry through conformal geometric algebra (a gaviewer tutorial). *available from http://www.science.uva.nl/ga*, 2003.
16. L. Dorst, D. Fontijne, and S. Mann. *Geometric Algebra for Computer Science, An Object-Oriented Approach to Geometry*. Morgan Kaufman, 2007.
17. L. Dorst and S. Mann. Geometric algebra: a computational framework for geometrical applications (part i: algebra). *Computer Graphics and Application*, 22(3):24–31, May/June 2002.
18. Leo Dorst. Honing geometric algebra for its use in the computer sciences. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*. Springer-Verlag, 2001.
19. J. Ebling. Clifford fourier transform on vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):469–479, 2005.
20. D. Fontijne, T. Bouma, and L. Dorst. Gaigen: A geometric algebra implementation generator. *Available at http://www.science.uva.nl/ga/gaigen*.
21. D. Fontijne and L. Dorst. Modeling 3D euclidean geometry. *IEEE Computer Graphics and Applications*, 23(2):68–78, MarchApril 2003.
22. D. Hestenes. *New Foundations for Classical Mechanics*. Dordrecht, 1986.
23. D. Hestenes and G. Sobczyk. *Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics*. Dordrecht, 1984.
24. D. Hildenbrand, D. Fontijne, Yusheng Wang, M. Alexa, and L. Dorst. Competitive runtime performance for inverse kinematics algorithms using conformal geometric algebra. In *Eurographics conference Vienna*, 2006.
25. D. Hildenbrand, H. Lange, Florian Stock, and Andreas Koch. Efficient inverse kinematics algorithm based on conformal geometric algebra using reconfigurable hardware. In *GRAPP conference Madeira*, 2008.
26. D. Hildenbrand and Joachim Pitt. The Gaalop home page. HTML document http://www.gaalop.de, 2008.
27. A. Lasenby J. Lasenby, E. Bayro-Corrochano and G. Sommer. A new methodology for computing invariants in computer vision. In *Proceedings of ICPR 96*, 1996.
28. Joan Lasenby, W. J. Fitzgerald, A.N. Lasenby, and C.J.L. Doran. New geometric methods for computer vision: An application to structure and motion estimation. *International Journal of Computer Vision*, 3(26):191–213, 1998.
29. S. Mann and L. Dorst. Geometric algebra: a computational framework for geometrical applications (part ii: applications). *Computer Graphics and Application*, 22(4):58–67, July/August 2002.
30. S. Mann, L. Dorst, and T. Bouma. The making of GABLE, a geometric algebra learning environment in matlab. pages 491–511, 2001.
31. Naeve and A. Rockwood. Course 53 geometric algebra. In *Siggraph conference Los Angeles*, 2001.
32. C. Perwass. The CLU home page. HTML document http://www.clucalc.info, 2005.
33. C. Perwass and W. Förstner. Uncertain geometry with circles, spheres and conics. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties from Incomplete Data*, volume 31 of *Computational Imaging and Vision*, pages 23–41. Springer-Verlag, 2006.

34. C. Perwass, C. Gebken, and G. Sommer. Geometry and kinematics with uncertain data. In A. Leonardis, H. Bischof, and A. Pinz, editors, *9th European Conference on Computer Vision, ECCV 2006, May 2006, Graz, Austria*, number 3951 in LNCS, pages 225–237. Springer-Verlag, Berlin Heidelberg, 2006.
35. C. Perwass and G. Sommer. The inversion camera model. In *28. Symposium für Mustererkennung, DAGM 2006, Berlin, 12.-14.09.2006*. Springer-Verlag, Berlin, Heidelberg, 2006.
36. C.B.U. Perwass. *Applications of Geometric Algebra in Computer Vision*. PhD thesis, Cambridge Universtiy, 2000.
37. C.B.U. Perwass and J. Lasenby. A Geometric Analysis of the Trifocal Tensor. In R. Kakarala R. Klette, G. Gimel'farb, editor, *Image and Vision Computing New Zealand, IVCNZ'98, Proceedings*, pages 157–162. The University of Auckland, 1998.
38. C.B.U. Perwass and J. Lasenby. A Unified Description of Multiple View Geometry. In G. Sommer, editor, *Geometric Computing with Clifford Algebra*. Springer-Verlag, 2001.
39. Christian Perwass. *Geometric Algebra with Applications in Engineering*. Springer, 2009.
40. M. Pham, K. Tachibana, E. Hitzer, T. Yoshikawa, and T. Furuhashi. Classification and clustering of spatial patterns with geometric algebra. In *AGACSE conference Leipzig*, 2008.
41. B. Rosenhahn. *Pose Estimation Revisited*. PhD thesis, Inst. f. Informatik u. Prakt. Mathematik der Christian-Albrechts-Universität zu Kiel, 2003.
42. B. Rosenhahn and G. Sommer. Pose estimation in conformal geometric algebra. *Journal of Mathematical Imaging and Vision*, 22:27–70, 2005.
43. G. Sommer, editor. *Geometric Computing with Clifford Algebra*. Springer Verlag, 2001.
44. G. Sommer. Applications of geometric algebra in robot vision. In H. Li, P.J. Olver, and G. Sommer, editors, *Computer Algebra and Geometric Algebra with Applications*, volume 3519 of *LNCS*, pages 258–277. 6th International Workshop IWMM 2004, Shanghai, China and International Workshop GIAE 2004, Xian, China, Springer-Verlag, Berlin Heidelberg, 2005.
45. G. Sommer, B. Rosenhahn, and C. Perwass. The twist representation of free-form objects. In R. Klette, R. Kozera, L. Noakes, and J. Weickert, editors, *Geometric Properties from Incomplete Data*, volume 31 of *Computational Imaging and Vision*, pages 3–22. Springer-Verlag, 2006.
46. Rich Wareham, Jonathan Cameron, and Joan Lasenby. Applications of conformal geometric algebra in computer vision and graphics. *Lecture Notes in Computer Science*, 3519:329–349, June 2005.
47. Rich Wareham and Joan Lasenby. Applications of conformal geometric algebra in computer vision and graphics. *submitted to ACM Transactions on Graphics*, 2004.
48. M. Zaharia and L. Dorst. Modeling and visualization of 3d polygonal mesh surfaces using geometric algebra. *Computers & Graphics*, 29(5):802–810, 2003.